New Vtx search routines

- Description of existing code & problems
- Description of new code
  - Vtxsearch2
  - Vtxsearch3
- Error guesstimate
- Comparison with already located events

Existing code: vtxsearch

- vtxsearch steps through a grid of U, V (1 mm) and Z (3 mm) vertex positions
- makes U lines and V lines with vertex constraint
- Finds a weight = sum of U and V line hits
- Finds position where weight is maximum
- Problems
  - *No UV correlation*
  - *No provision for preserving good tracks*

vtxsearch2

- Allows user to call vtxsearch3 if a high momentum track with SFT hits exists
- Uses the same grid pattern as vtxsearch
- Finds U view SFT lines using vertex constraint
- Call to sftk_uvx
  - Finds X view lines if 2 or more X planes in station
  - Makes V view lines using U and X views
  - Calculates a weight = Σ(U line hits) * Σ(V line hits)
- Finds position where weight is maximum – “predicted vertex position”
- Calculates a vertex error – more on this later

vtxsearch3

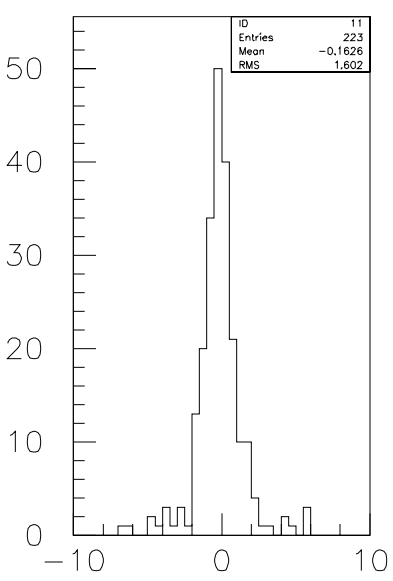
- Searches for a vertex position along a track direction
- Entire target is scanned in Z steps of 3 mm along projected track direction
- Find vtx position using weight from sftk_uvx
- Vertex Z error from sftk_uvx
- Vertex U,V errors from track fit w/o vertex constraint

Error Guesstimate

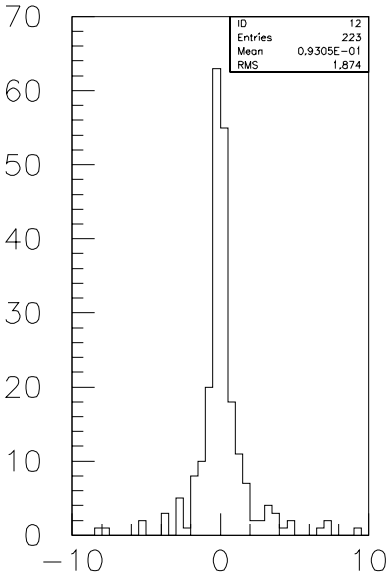
- Assume the vertex weight has a gaussian shape
- Step away from the predicted vertex position in +/- U,V,(Z) in 1(3) mm steps until weight falls to 68% of the maximum

Check code using located events

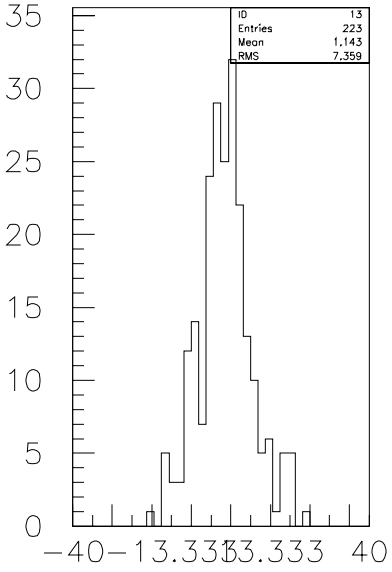
- Used 223 located events
- Moved the located vertex position randomly within +/- 5 mm in U,V and +/- 10 mm in Z
- Call vtxsearch2 to get predicted vertex position and errors
- Attached plots
  - Top row: dvx U = $U_{loc} - U_{pred}$ (mm), etc
  - Bottom row: dvx normalized by calculated error
- *U,V errors are OK*
- *Z error over-estimated by 50%*



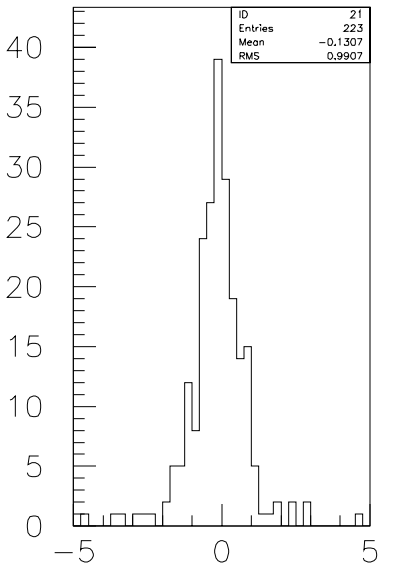
dvx U



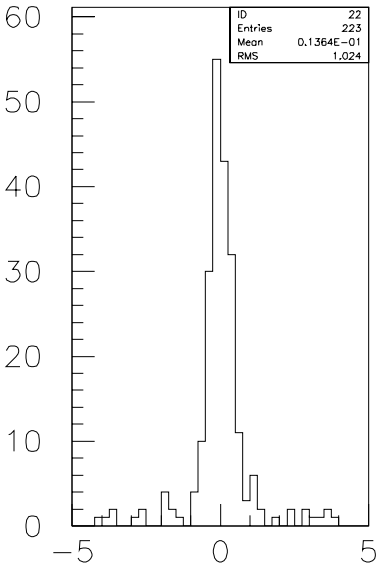
dvx V



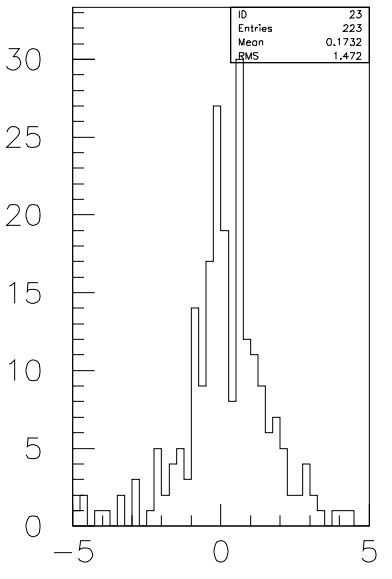
dvx Z



ndvx U



ndvx V



ndvx Z